What is claimed is:

1. (Currently Amended) A method of determining whether materialized views and indexes relate to a query, said method comprising:

determining if an XPath expression within a query is associated with an XPath expression pre-computed by an index or materialized view;

computing an XPath expression to be sent to said index or materialized view;

identifying indexes or materialized views that match said XPath expression;

computing compensation for index selection or materialized view matching to determine what portion of said query can be evaluated by said index; and

calculating an XPath predicate to be applied as a residual query, wherein said residual query comprises at least a portion of said query not associated with said index or said materialized view.

2. (Original) The method in claim 1, wherein said process of determining whether said XPath expression within said query is associated with said index or materialized view comprises representing said XPath expression as a tree of XPath steps.

3. (Original) The method in claim 2, wherein each of said XPath steps comprise an XPath step node comprising axis data, test data, predicate data, and next XPath step node data.

4. (Original) The method in claim 2, wherein said process of determining whether said XPath expression within said query is associated with said index or materialized view comprises

detecting containment mappings between XPath expressions in said query and said index or materialized view.

5. (Original) The method in claim 4, wherein said process of detecting containment mappings comprises traversing said tree of XPath steps from the top down and, during said traversing, matching axis data, test data, predicate data, and next XPath step node data with index or materialized view XPath expressions.

6. (Original) The method in claim 5, further comprising before said matching process, moving predicate conditions into filter expressions.

7. (Original) The method in claim 2, wherein conjunctions and disjunctions are permitted in said tree of XPath steps if:

at least one disjunction in said materialized view exists within said query; and

said materialized view maps to at least one expression other than the disjunct of the query.

8. (Canceled).

9. (Original) The method in claim 8, wherein said process of computing said compensation comprises determining what portion of said query is not contained within said index.

10. (Original) The method in claim 1, further comprising applying said query to documents within indexes or materialized views that match said XPath expression.

11. (Currently Amended) A method of determining whether materialized views and indexes relate to a query, said method comprising:

determining if an XPath expression within a query is associated with an XPath expression within an index or materialized view by representing said XPath expression as a tree of XPath steps, wherein each of said XPath steps comprise an XPath step node comprising axis data, test data, predicate data, and next XPath step node data;

computing an XPath expression to be sent to said index or materialized view;

identifying indexes or materialized view that match said XPath expression;

computing compensation for index selection or materialized view matching to determine what portion of said query can be evaluated by said index; and

calculating an XPath predicate to be applied as a residual query, wherein said residual query comprises at least a portion of said query not associated with said index or said materialized view.

12. (Original) The method in claim 11, wherein said process of determining whether said XPath expression within said query is associated with said index or materialized view comprises detecting containment mappings between XPath expressions in said query and said index or materialized view.

13. (Original) The method in claim 12, wherein said process of detecting containment mappings comprises traversing said tree of XPath steps from the top down and, during said traversing, matching axis data, test data, predicate data, and next XPath step node data with index or materialized view XPath expressions.

14. (Original) The method in claim 13, further comprising before said matching process, moving predicate conditions into filter expressions.

15. (Original) The method in claim 11, wherein conjunctions and disjunctions are permitted in said tree of XPath steps if:

at least one disjunction in said materialized view exists within said query; and

said materialized view maps to at least one node other than the disjunct of the query.

16. (Canceled).

17. (Original) The method in claim 16, wherein said process of computing said compensation comprises determining what portion of said query is not contained within said index.

18. (Original) The method in claim 11, further comprising applying said query to documents within indexes or materialized views that match said XPath expression.

19. (Currently Amended) A method of determining whether materialized views and indexes

relate to a query, said method comprising:

determining if an XPath expression within a query is associated with an XPath expression within an index or materialized view;

computing an XPath expression to be sent to said index or materialized view;

identifying indexes or materialized view that match said XPath expression;

computing compensation for index selection or materialized view matching to determine what portion of said query can be evaluated by said index;

calculating an XPath predicate to be applied as a residual query, wherein said residual query comprises at least a portion of said query not associated with said index or said materialized view; and

computing compensation for index selection or materialized view matching to determine what portion of said query can be evaluated by said index so as to determine what portion of said query is not contained within said index.

20. (Original) The method in claim 19, wherein said process of determining whether said XPath expression within said query is associated with said index or said materialized view comprises representing said XPath expression as a tree of XPath steps.

21. (Original) The method in claim 20, wherein each of said XPath steps comprise an XPath step node comprising axis data, test data, predicate data, and next XPath step node data.

22. (Original) The method in claim 21, wherein said process of determining whether said

XPath expression within said query is associated with said index or said materialized view comprises detecting containment mappings between XPath expressions in said query and said index or materialized view.

23. (Original) The method in claim 22, wherein said process of detecting containment mappings comprises traversing said tree of XPath steps from the top down and, during said traversing, matching axis data, test data, predicate data, and next XPath step node data with index or materialized view XPath expressions.

24. (Original) The method in claim 24, further comprising before said matching process, moving predicate conditions into filter expressions.

25. (Original) The method in claim 20, wherein conjunctions and disjunctions are permitted in said tree of XPath steps if:

at least one disjunction in said materialized view exists within said query; and

said materialized view maps to at least one node other than the disjunct of the query.

26. (Original) The method in claim 19, further comprising applying said query to documents within indexes or materialized views that match said XPath expression.

27. (Currently Amended) A service for determining whether materialized views and indexes relate to a query, said service comprising:

determining if an XPath expression within a query is associated with an XPath expression pre-computed by an index or materialized view;

computing an XPath expression to be sent to said index or materialized view;

identifying indexes or materialized views that match said XPath expression;

computing compensation for index selection or materialized view matching to determine what portion of said query can be evaluated by said index; and

calculating an XPath predicate to be applied as a residual query, wherein said residual query comprises at least a portion of said query not associated with said index or said materialized view.

28. (Original) The service in claim 27, wherein said process determining whether said XPath expression within said query is associated with said index or materialized view comprises representing said XPath expression as a tree of XPath steps.

29. (Original) The service in claim 28, wherein each of said XPath steps comprise an XPath step node comprising axis data, test data, predicate data, and next XPath step node data.

30. (Original) The service in claim 28, wherein said process of determining whether said XPath expression within said query is associated with said index or materialized view comprises detecting containment mappings between XPath expressions in said query and said index or materialized view.

31. (Original) The service in claim 30, wherein said process of detecting containment mappings comprises traversing said tree of XPath steps from the top down and, during said traversing, matching axis data, test data, predicate data, and next XPath step node data with index or materialized view XPath expressions.

32. (Original) The service in claim 31, further comprising, before said matching process, moving predicate conditions into filter expressions.

33. (Original) The service in claim 28, wherein conjunctions and disjunctions are permitted in said tree of XPath steps if:

at least one disjunction in said materialized view exists within said query; and

said materialized view maps to at least one expression other than the disjunct of the query.

34. (Canceled).

35. (Original) The service in claim 34, wherein said process of computing said compensation comprises determining what portion of said query is not contained within said index.

36. (Original) The service in claim 27, further comprising applying said query to documents within indexes or materialized views that match said XPath expression.

37. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform a method of extracting circuit characteristics from a circuit design, said method comprising determining whether materialized views and indexes relate to a query, said method comprising:

determining if an XPath expression within a query is associated with an XPath expression pre-computed by an index or materialized view;

computing an XPath expression to be sent to said index or materialized view;

identifying indexes or materialized views that match said XPath expression;

computing compensation for index selection or materialized view matching to determine what portion of said query can be evaluated by said index; and

calculating an XPath predicate to be applied as a residual query, wherein said residual query comprises at least a portion of said query not associated with said index or said materialized view.

38. (Original) The program storage device in claim 37, wherein said process of determining whether said XPath expression within said query is associated with said index or materialized view comprises representing said XPath expression as a tree of XPath steps.

39. (Original) The program storage device in claim 38, wherein each of said XPath steps comprise an XPath step node comprising axis data, test data, predicate data, and next XPath step node data.

40. (Original) The program storage device in claim 39, wherein said process of determining whether said XPath expression within said query is associated with said index or materialized view comprises detecting containment mappings between XPath expressions in said query and said index or materialized view.

41. (Original) The program storage device in claim 40, wherein said process of detecting containment mappings comprises traversing said tree of XPath steps from the top down and, during said traversing, matching axis data, test data, predicate data, and next XPath step node data with index or materialized view XPath expressions.

42. (Original) The program storage device in claim 41, wherein said method further comprises before said matching process, moving predicate conditions into filter expressions.

43. (Original) The program storage device in claim 38, wherein conjunctions and disjunctions are permitted in said tree of XPath steps if:

at least one disjunction in said materialized view exists within said query; and

said materialized view maps to at least one expression other than the disjunct of the query.

44. (Canceled).

45. (Original) The program storage device in claim 44, wherein said process of computing

said compensation comprises determining what portion of said query is not contained within said index.

46. (Original) The program storage device in claim 37, wherein said method further comprises applying said query to documents within indexes or materialized views that match said XPath expression.

47. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 43 wherein the auxiliary structures include a
number of indexes, a number of partial XML indexes, and a number of materialized views.

48. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 43 wherein the pre-computed information includes
pre-computed XPath results (PXRs).

49. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 43 wherein the user query processing further
comprises navigating path expressions with a query language.

50. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 49 wherein the query language employs XPath.

51. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 49 wherein the query language includes at least one of: XQuery, SQL/XML, and XSLT.

52. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 43 wherein the detecting further comprises:

selectively executing a set of predetermined sequential rules for traversing a tree of nodes;

matching node data with the pre-computed information; and

selecting auxiliary structures that subsume portions of the user query.

53. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 52 wherein the node data includes axis data, test data, predicate data, and next step node data.

54. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 52 further comprising normalizing expression trees by moving predicate conditions into filter expressions before the identifying.

55. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 43 wherein executing the query further comprises:

constructing a pushdown expression for evaluation with information in the auxiliary structure; and

constructing a compensation expression for evaluation as ~~a~~ said residual query, wherein said residual query comprises at least a portion of said query not associated with said index or said materialized view.

56. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 55 wherein the compensation expression is an
XPath predicate.

57. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 55 further comprising building a taxonomy of
auxiliary structures.

58. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 57 further comprising classifying compensation
expressions for the taxonomy according to a predetermined set of values.

59. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 43 wherein the identifying handles at least one of:

nested path expressions, nested predicates, value-based comparison predicates, conjunction, disjunction, all XPath axes, branches, and wild cards.

60. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 59 wherein the XPath axes include child,
descendant, self, attribute, parent, and descendant-or-self.

61. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 43 further comprising creating a mapping directed
acyclic graph (DAG) that separately encodes a set of all containment mappings for each node.

62. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 61 wherein creating the mapping DAG is
polynomial in terms of ~~the~~ a size of ~~the~~ expression trees.

63. (Currently Amended) The ~~computer~~ program storage device ~~product~~ of claim 61 further comprising pruning the mapping DAG to
remove invalid node pairs.

64. (Currently Amended) A system for querying a structured document, comprising:

means for identifying auxiliary structures including pre-computed information applicable to accelerate user query processing by detecting containment mappings between query expressions and expressions in the auxiliary structures; and means for finding the user query result by executing a residual query that exploits the pre-computed information for each detected containment mapping, wherein said residual query comprises at least a portion of said query not associated with said index or said materialized view.